Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended):   A method comprising:

merging ~~at least~~ two sections, each including a header, from an object file into one new section of a firmware module, wherein (a) the new section includes data, executable code, and no more than a single header and (b) the firmware module follows a portable executable (PE) format having subdivisions that include an MS-DOS header;

storing the firmware module in memory, wherein the stored module includes the new section, another section including relocation information, and no additional section; and

flattening the firmware module by replacing existing content within at least one field within the MS-DOS header of the firmware module with fill data that is more compressible than the existing content.

Claim 2 (canceled)

Claim 3 (original):   A method according to claim 1, wherein the operation of flattening the firmware module comprises loading fill data into an MS-DOS stub field within the MS-DOS header.

Claim 4 (original):   A method according to claim 1, wherein the operation of flattening the firmware module comprises ensuring that fill data occupies all fields within the MS-DOS header except for an lfanew field and an e-magic field.

Claim 5 (original):   A method according to claim 1, wherein the PE format also includes an optional file header, the method further comprising: loading fill data into at least one field within the optional file header.

Claim 6 (original):   A method according to claim 5, wherein the operation of loading fill data into at least one field within the optional file header comprises:

loading fill data into at least one of a SizeOfStackReserve field, a SizeOfStackCommit field, a SizeOfHeapReserve field, a SizeOfHeapCommit field, and a LoaderFlags field.

Claim 7 (canceled)

Claim 8 (previously presented):   A method according to claim 1, wherein the operation of merging at least two sections from an object file into one section in the firmware module comprises

instructing a linker to merge the at least two sections when generating the firmware module from the object file.

Claim 9 (original):   A method according to claim 8, further comprising: causing the linker to change a name of a section specified in the object file to a more compressible name.

Claim 10 (original):   A method according to claim 1, wherein the PE format also includes an image page, the method further comprising:

storing, in the image page, an alternate file path for a debug file associated with the firmware module, wherein the alternate file path is more compressible than an original file path for the debug file.

Claim 11 (original):   A method according to claim 1, wherein the PE format also includes an image page, the method further comprising:

instructing a linker to store, in the image page of the firmware module, an alternate file path for a debug file associated with the firmware module, wherein the alternate file path is more compressible than an original file path for the debug file.

Claim 12 (currently amended):   A program product comprising:

a machine accessible storage medium; and

instructions encoded in the machine accessible medium, wherein the instructions, when executed by a processing system, cause the processing system to perform operations comprising:

3

merging ~~at least~~ two sections, each including a header, from an object file into one new section of a firmware module, wherein (a) the new section includes data, executable code, and no more than a single header and (b) the firmware module follows a portable executable (PE) format having subdivisions that include an MS-DOS header;

accessing the firmware module within the processing system; and

flattening the firmware module by replacing existing content within at least one field within the MS-DOS header of the firmware module with fill data that is more compressible than the existing content.

Claim 13 (canceled)

Claim 14 (original):   A program product according to claim 12, wherein the operation of flattening the firmware module comprises loading fill data into an MS-DOS stub field within the MS-DOS header.

Claim 15 (original):   A program product according to claim 12, wherein the operation of flattening the firmware module comprises ensuring that fill data occupies all fields within the MS-DOS header except for an lfanew field and an e-magic field.

Claim 16 (original):   A program product according to claim 12, wherein the PE format also includes an optional file header, the program product further comprising:

instructions which, when executed by the processing system, cause the processing system to load fill data into at least one field within the optional file header.

Claim 17 (original):   A program product according to claim 16, wherein the operation of loading fill data into at least one field within the optional file header comprises:

loading fill data into at least one of a SizeOfStackReserve field, a SizeOfStackCommit field, a SizeOfHeapReserve field, a SizeOfHeapCommit field, and a LoaderFlags field.

Claim 18 (currently amended):   A processing system with resources for flattening a firmware module, the processing system comprising:

a processor;

memory communicatively coupled to the processor; and

instructions stored in the memory, wherein the instructions, when executed by the processor, cause the processing system to perform operations comprising:

merging at least two sections, each including a header, from an object file into one new section of a firmware module, wherein (a) the new section includes data, executable code, and no more than a single header and (b) the firmware module follows a portable executable (PE) format having subdivisions that include an MS-DOS header;

accessing the firmware module within the processing system; and

flattening the firmware module by replacing existing content within at least one field within the MS-DOS header of the firmware module with fill data that is more compressible than the existing content.

Claim 19 (original):   A processing system according to claim 18, wherein the operation of flattening the firmware module comprises loading fill data into at least fifty bytes of the MS-DOS header.

Claim 20 (original):   A processing system according to claim 18, wherein the operation of flattening the firmware module comprises loading fill data into an MS-DOS stub field within the MS-DOS header.

Claim 21 (original):   A processing system according to claim 18, wherein the operation of flattening the firmware module comprises ensuring that fill data occupies all fields within the MS-DOS header except for an lfanew field and an e-magic field.

Claims 22 - 23 (canceled)

Claim 24 (currently amended):   An apparatus comprising:

a machine accessible storage medium; and

a firmware module encoded in the machine accessible medium, the firmware module having a portable executable (PE) format with subdivisions that include an MS-DOS header, wherein the firmware module was produced by operations comprising:

merging at least two sections, each including a header, from an object file into one new section of a firmware module, wherein the new section includes data, executable code, and no more than a single header; and

flattening the firmware module by replacing existing content within at least one field within the MS-DOS header of the firmware module with fill data that is more compressible than the existing content.

Claim 25 (original):   An apparatus according to claim 24, further comprising:

a processor communicatively coupled to the machine accessible medium;

memory communicatively coupled to the processor; and

instructions stored in the memory, wherein the instructions, when executed by the processor, cause the processing system to perform operations comprising:

retrieving the firmware module from the machine accessible medium; and

executing the firmware module within a boot environment.

Claim 26 (original):   An apparatus according to claim 24, wherein:

the machine accessible medium comprises a non-volatile storage device; and

the apparatus further comprises an interface in communication with the non-volatile storage device, the interface operable to provide communication between the non-volatile storage device and a processor of a data processing system.

Claim 27 (original):   An apparatus according to claim 26, wherein the apparatus comprises an adapter card for a processing system.

Claim 28 (new):   A method according to claim 1, wherein the unmodified firmware module includes material necessary for the module to properly execute and perform a specific function, and the method further includes flattening the material to produce a modified firmware module still configured to properly execute and perform the specific function.

Claim 29 (new):   A method according to claim 28, wherein the flattened material is configured to be executed during boot services.

Claim 30 (new):   A method according to claim 1 including using a post-build tool to flatten a field included in a Common Object File Format (COFF) file header; wherein flattening includes modifying content included in the COFF file header to increase the compressibility of the COFF file header.

Claim 31 (new):   A method according to claim 1 including merging .text, .data, and .rdata sections into the new section and naming the new section with a single character.